

A Covert Queueing Channel in Round Robin Schedulers

AmirEmad Ghassami*, Ali Yekkehkhany*, Negar Kiyavash*[†], Yi Lu*
 *Department of ECE, [†]Department of ISE, and *Coordinated Science Laboratory
 University of Illinois at Urbana-Champaign, Urbana, Illinois 61801
 Email: {ghassam2,yekkehk2,kiyavash,yilu4}@illinois.edu

Abstract—We study a covert queueing channel between two users sharing a round robin scheduler. Such a covert channel can arise when users share a resource such as a computer processor or a router arbitrated by a round robin policy. We present an information-theoretic framework to model and derive the maximum reliable data transmission rate, i.e., the capacity of this channel for both noiseless and noisy scenarios. Our results show that seemingly isolated users can communicate with high rate over the covert channel. Furthermore, we propose a practical finite-length code construction, which achieves the capacity limit.

Index Terms—Covert Queueing Channel, Round Robin Scheduler, Capacity Limit.

I. INTRODUCTION

Shared resources between users in a system can lead to occurrence of covert and side channels. Such channels indirectly connect users without the designer's intent [1]–[10]. In covert channel of our interest, users utilize the coupling through a shared resource to communicate with each other. Covert channels have traditionally been used by trusted insiders or a malware who have access to secret information to leak it to untrusted outsiders [11], [12]. To efficiently utilize the covert channel, users must agree on a usage pattern to optimally take advantage of the features of the shared resource. In contrast, in a side channel a malicious user attempts to gain access to another user's private information while there is no collaboration between the users [13]. More specifically, queueing side channels, which are special kinds of side channels, have been studied in the literature [6], [7].

Multiple users running on a computer who are using hardware resources such as CPU, storage, and multiple network streams flowing through a common router are examples of environments in which covert and side channels can be created.

The focus of this work is on covert queueing channel, a special kind of covert channel, that appears as a result of users sharing a job scheduler. In such channels, information is transmitted between the users through the delays experienced by one of the users because the scheduler is busy serving the other user. More specifically, due to the inter-dependencies between delays observed by users, if one user experiences delays in his service, he understands that the other user was issuing jobs. Different schedulers such as First-Come-First-Served (FCFS), Time-Division-Multiple-Access (TDMA), round robin, etc. can be used for resource sharing. Clearly the optimal scheme for message transmission

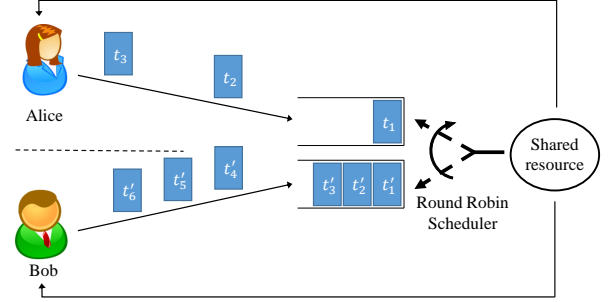


Fig. 1: System Setup.

and the rate of communication between the users depends on the scheduling policy of the shared resource.

For each scheduling policy, there is a tradeoff between throughput and security. For throughput, as long as the rates at which users request the shared resource is within the system's capacity region, an effective scheduler should be able to respond to the users' requests in a stable fashion. Such a scheduler is called a throughput optimal scheduler. It has been shown that from the security viewpoint, TDMA is the most secure scheduling policy [7]. In this type of scheduler, since the serving times of the users are decoupled in time, users' delays are independent of each other and hence, no information can be conveyed through the scheduler. However, the decoupling can cause significant delays in service given to users. Hence, TDMA is not throughput optimal. The covert queueing channel created among users when the scheduler is FCFS is studied in details in [11]. Although this scheduler does not waste any resource and hence is throughput optimal, it allows users to communicate with an information rate as high as 0.8114 bits per time slot.

In this paper we focus on round robin scheduler, which is another throughput optimal policy commonly used in computer processors and communication networks. Kadloor et al. [14] showed that when dealing with queueing side channels, round robin scheduling policy is privacy optimal within the class of work-conserving policies. In this work, we focus on covert channels created in this scheduler. We show that users can communicate with an information rate of 0.6942 bits per time slot through the covert channel created between them in this system in the absence of noise. Additionally, we study the noisy version of this covert channel in which packets are dropped with a certain probability, and we compute the

capacity as a function of packet drop probability.

Followings are the main contributions of this work. The optimum signaling scheme for the covert queuing channel with round robin scheduler has been characterized (Section III), and the capacity of this covert queuing channel is shown to be approximately 0.6942 bits per time slot (Subsection IV-A). Furthermore, a practical optimal finite block length coding scheme is proposed, both when codewords are of fixed and of variable length. Our results show that the rates of the proposed optimal coding schemes approach the capacity as the number of messages goes to infinity (Subsection IV-B). Finally, the model is extended to a more realistic noisy case, and the capacity is calculated (Section V).

II. SYSTEM MODEL

We consider the system depicted in Figure 1, in which a shared resource services jobs from two users, Alice and Bob, using round robin policy. In this depiction, each packet is marked by its arrival time. As shown in this figure, there is a feedback line from the shared resource to the users, which notifies them when their packet is served. Clearly, this allows the users to infer the status of the head of their queue.

Time is assumed to be discretized into slots, and each packet generated by users is served in one time slot. The scheduler can serve one packet in each time slot. We follow the common convention that the packets arrive at the beginning of time slots and the departures occur at the end of time slots. Each user's packets are buffered in a separate queue, and the round robin scheduler picks packets from the two queues as follows. In each time slot, three cases may happen: (a) If both users' arrival queues are empty, the system remains idle and resumes scheduling in the next time slot. (b) If only one user's queue has a packet, the current slot is given to that user, and the scheduler continues scheduling in the next time slot. (c) If both users have waiting packets, the scheduler always gives priority to a fixed user. That is, the current time slot is allocated to serve a packet from the user with higher priority, and the next time slot will be allocated to the other user. The system continues scheduling after both users have received service. Without loss of generality, we assume that the priority is always given to Bob in the sequel.

We assume both Alice and Bob send at most one packet per time slot. Thus, their packet stream can be modeled as a binary bit stream, where bit '1' indicates a packet was sent, and bit '0' indicates no packet was transmitted. Since the scheduler can serve at most one packet per time slot, the sum of users' packet rates should be less than one for stability.

Figure 2 depicts an example of the system scheduling. In this and other such figures, Alice's and Bob's packets are shown by circled tip and regular arrows, respectively. For each user, the arrival stream, the head-of-the-queue stream and the departure stream are shown. Here, arrival stream is the actual packet stream sent by the user, and head-of-the-queue stream is the packets ready to be served at the head of the corresponding user's queue. Therefore, at any given time slot, the head of the queue can be '1' even though no packet has arrived in that

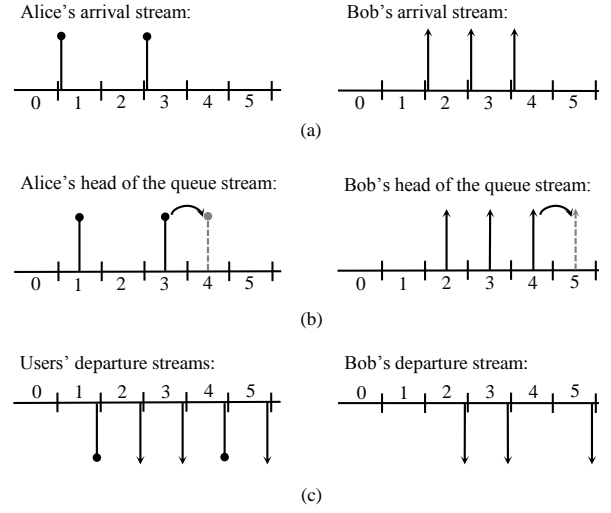


Fig. 2: (a) Arrival streams; (b) head-of-the-queue streams; (c) departure streams.

slot. In Figure 2(b), the packet denoted by the gray dashed line indicates that it has been the same as its previous packet, which has been made to wait in the queue for one time slot to receive service in the next time slot (we emphasize that the gray dashed symbols are not packet arrival). The downward streams (Figure 2 (c)) indicate the departure time of users' packets.

Suppose Alice aims to send message W uniformly drawn from the set $\{1, 2, \dots, M\}$. To this end, Alice encodes this message to a bit stream X^m which is sent out as a packet stream with arrival times A_A^n . Based on scheduling policy and both Alice's and Bob's packet arrivals, Bob receives a stream of acknowledgements from the system which is denoted by D_B^n . Finally, Bob transforms this stream to a bit stream Y^m which will be decoded to message \hat{W} . As a result, we have the following Markov chain:

$$W \rightarrow X^m \rightarrow A_A^n \rightarrow D_B^n \rightarrow Y^m \rightarrow \hat{W} \quad (1)$$

The noise in the system is modeled as follows. The packets generated by either Alice or Bob may be dropped in the link between the users and the shared resource with probability δ . Note that this noise can affect the transmissions in $X^m \rightarrow A_A^n$ and $A_A^n \rightarrow D_B^n$ in Markov chain (1). We will later show that with the optimum signaling scheme between Alice and Bob, noise will not affect the process $A_A^n \rightarrow D_B^n$.

III. OPTIMUM SIGNALING SCHEME

The main idea behind the signaling scheme from Alice to Bob is to utilize the delays occurred in Bob's departure stream caused by Alice's packets. First, we investigate the optimum stream sent by Bob.

Lemma 1. *In order that Bob maximizes his inference from Alice's signaling (the number of time slots in which Bob receives a bit from Alice), he should have a ready-to-be-served packet at the head of his queue in all time slots.*

Proof: Proof by contradiction. Suppose Bob does not have any packets to be served at time slot n . Then the round robin

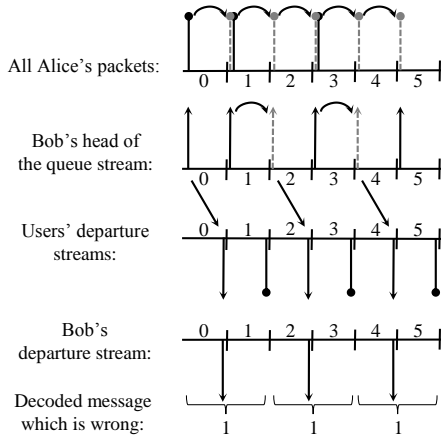


Fig. 3: Visualization of Remark 1.

policy will restart scheduling in time slot $n + 1$ regardless of whether Alice had a packet in time slot n or not. Therefore, Alice cannot affect the departure time of Bob. This means that Alice cannot communicate with Bob in time slot n . ■

The requirement that Bob should have a ready-to-be-served packet at all time slots does not mean that he has to send a packet in all time slots. It suffices for him to fix his queue length at some nonzero length, and whenever one of his packets is served, he generates a packet to ensure his queue length remains nonzero. This strategy allows him to keep the sum rate of arrivals from Alice and Bob less than 1 and keep the system stable. In the following we illustrate the effect of Alice's bits '0' and '1' on the acknowledgments given to Bob. **Signaling bit '1':** To signal bit '1' in time slot n , Alice must have a head-of-the-queue packet at the beginning of the time slot. Recall that Bob has a ready-to-be-served packet in all time slots. Thus, round robin policy will serve Bob and Alice at time slots n and $n + 1$, respectively. Therefore, when Bob receives service in a time slot but does not receive service in the next time slot, he decodes bit '1'.

Signaling bit '0': To signal bit '0' in time slot n , Alice must not have a head-of-the-queue packet at the beginning of the time slot. Because Bob has a packet which is ready to be serviced in the head of the queue in this time slot, he receives service at time slot n , and the scheduler resets for time slot $n + 1$. As a result, at time slot $n + 1$, Bob is served again. Therefore, if Bob receives service in two consecutive time slots, he decodes it as bit '0'.

Remark 1. Note that Alice should not send two packets in two consecutive time slots. This is because if Alice sends two (or more) packets in consecutive time slots, her next (or more) '0'(s) would disappear as her packets are accumulated in the queue, as depicted in Figure 3. Hence, a bit '1' in a message effectively requires two time slots for transmission. Therefore, Alice must idle for one time slot after she sends a packet. See Figure 4 for an example of an effective communication, in which Alice is sending the bit stream 1101001 to Bob.

In the next section, we find the maximum achievable rate at which Alice can communicate reliably with Bob.

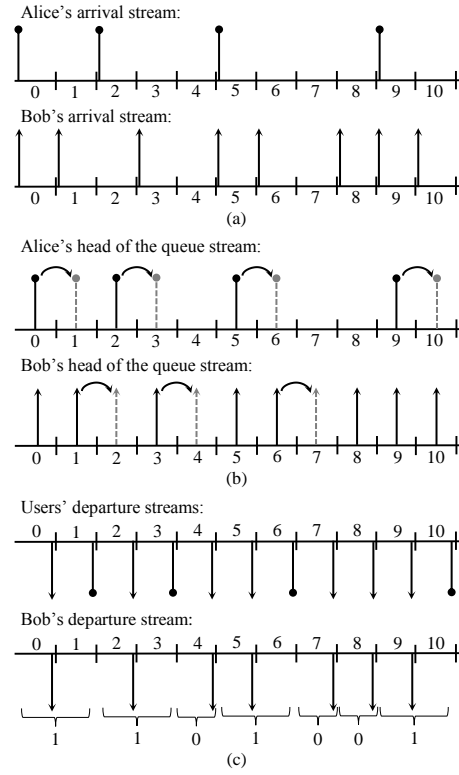


Fig. 4: Example of correct signaling between Alice and Bob.

IV. NOISELESS COVERT CHANNEL

In this section we calculate the capacity of the introduced covert channel and investigate the optimum coding schemes in finite-length codeword regime.

A. Coding Theorem

First, we define our performance metric, code, information transmission rate of a code, achievable rate, and channel capacity. These definitions are natural extensions of the classical definitions in information theory [1]. The used performance metric is the average error probability, that is $P_e \triangleq \mathbb{P}(W \neq \hat{W}) = \sum_{m=1}^M \frac{1}{M} \mathbb{P}(\hat{W} \neq m | W = m)$.

Definition 1. An (n, M, ϵ) -code consists of a codebook of size M with equiprobable binary codewords of average length n satisfying $P_e \leq \epsilon$.

Definition 2. The information transmission rate of a code is $R = \frac{\log M}{n}$, which is the amount of conveyed information normalized by the average number of used time slots (throughout the paper, all the logarithms are in base 2).

Definition 3. A rate R is said to be achievable if there exists a sequence of (n, M, ϵ_n) -codes such that $\epsilon_n \rightarrow 0$ as $n \rightarrow \infty$.

Definition 4. The channel capacity is the supremum achievable rate at which Alice can communicate through the covert channel with Bob.

Theorem 1. The capacity of the introduced covert channel between Alice and Bob created by the shared resource arbitrated by a round robin scheduler is

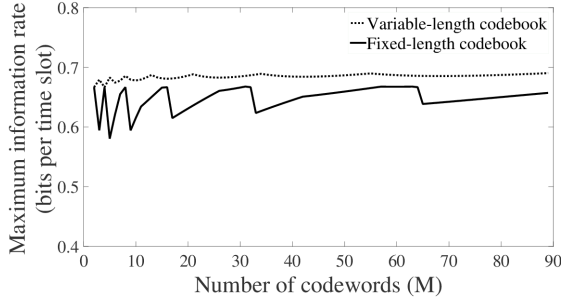


Fig. 5: Maximum information rate.

$$C = \sup_{p \in [0,1]} \frac{h(p)}{1+p}, \quad (2)$$

where p is the probability of sending message bit '1' by Alice and $h(\cdot)$ is the binary entropy function. The maximum of (2) is approximately 0.6942 achieved at $p = \frac{3-\sqrt{5}}{2}$.

Theorem 1 is a special case of Theorem 2 and hence, we present the general proof for Theorem 2.

B. Finite-length Codeword Regime

As mentioned earlier, Alice encodes each message to a binary sequence and creates a codebook \mathcal{C} , known to both Alice and Bob. The codewords in the codebook could be all of the same or different lengths. In the following, we briefly mention our proposed schemes for finding the optimum codebook for both scenarios. The details are omitted due to space constraints and could be found in [15].

1) *Variable-length Codewords*: By Definition 2, we have $R = \frac{M \log(M)}{2n_1 + n_0}$, where n_0 and n_1 are the number of bits '0' and '1' in the codebook, respectively. Given that M is the fixed given parameter, maximizing the rate is equivalent to searching for a codebook which achieves the minimum of the denominator in the above expression for R . Algorithm 1 describes how the M optimal codewords are obtained.

Algorithm 1 Optimum codebook with M codewords

- 1: Initialize the codebook to be $\{0, 1\}$
 - 2: **while** number of codewords is less than M **do**
 - 3: Choose a codeword with the minimum cost in the current codebook. Replace it by two new codewords by adding '0' and '1' to the right side of the old codeword, to build a new codebook.
 - 4: **end while**
-

In [15] we have proven that for a fixed given number of equiprobable messages, Algorithm 1 is optimal in the sense that it provides a codebook which maximizes the communication rate between the users. Also we have proven that the information transmission rate of a codebook created by this Algorithm converges to the capacity of the covert channel as the number of messages goes to infinity. The maximum rate at which Alice can communicate with Bob versus the number of codewords, M , is depicted in Figure 5.

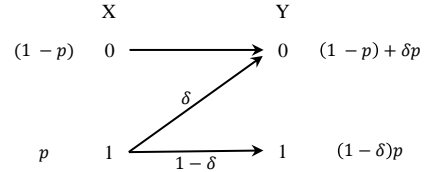


Fig. 6: Z-channel model of the covert channel with packet drop.

2) *Fixed-length Codewords*: In many applications, using variable-length codewords is not desirable from the designer's point of view. For example, in a noisy system, a variable-length scheme may lead to loss of synchronization between encoder and decoder. Denote the cost of a codebook with $\eta(\mathcal{C})$. Note that $\eta(\mathcal{C}) = n_0(\mathcal{C}) + 2n_1(\mathcal{C})$.

Algorithm 2 Optimum fixed-length codebook with M codewords

- 1: Set $\hat{l} = \lceil \log(M) \rceil$.
 - 2: **for** $l = \hat{l}$ to $2\hat{l}$ **do**
 - 3: $\mathcal{C}_l =$ Set of M codewords all of length 2^l with the least number of bits '0'.
 - 4: $\eta(\mathcal{C}_l) = n_0(\mathcal{C}_l) + 2n_1(\mathcal{C}_l)$.
 - 5: **end for**
 - 6: Output \mathcal{C}_{l^*} such that $l^* = \arg \min_l \eta(\mathcal{C}_l)$.
-

In [15] we have proven that for a fixed given number of equiprobable messages, Algorithm 2 outputs the optimal fixed-length codebook. Also we have proven that the information transmission rate of a codebook created by this Algorithm converges to the capacity of the covert channel as the number of messages goes to infinity (Figure 5).

V. NOISY COVERT CHANNEL

In this section we consider the case where the channel between the users is noisy. The noise model is as follows. We assume that a packet generated by a user is dropped with probability δ . In the following lemma, we investigate the effect of the noise on the covert channel between the users.

Lemma 2. In the optimum signaling scheme proposed in Section III, packet drop converts the channel between Alice and Bob to a Z-channel. That is, '0' is always transmitted error free, but '1' is flipped with probability δ (Figure 6).

Proof of Lemma 2 is omitted due to space constraint and could be found in [15].

Theorem 2. The capacity of the noisy covert channel between Alice and Bob with drop probability δ resulting from round robin scheduler is

$$C = \sup_{p \in [0,1]} \frac{h((1-\delta)p) - ph(\delta)}{(1-p) + \delta p + 2(1-\delta)p}, \quad (3)$$

where p is the probability of sending bit '1' by Alice and $h(\cdot)$ is the binary entropy function.

Proof: The proof consists of achievability and converse arguments.

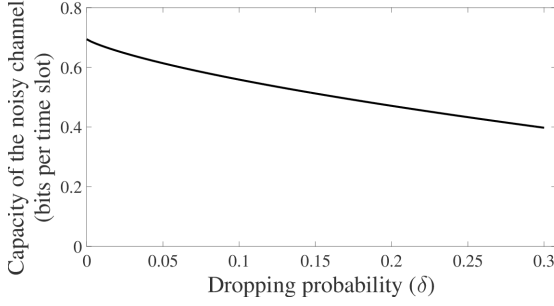


Fig. 7: Capacity of the noisy covert channel versus δ .

Converse: For any (n, M, ϵ) -code we have

$$\begin{aligned} \frac{1}{n} \log M &= \frac{1}{n} H(W) = \frac{1}{n} I(W; \hat{W}) + \frac{1}{n} H(W|\hat{W}) \\ &\stackrel{(a)}{\leq} \frac{1}{n} I(W; \hat{W}) + \epsilon_n \leq \frac{1}{n} I(X^m; Y^m) + \epsilon_n, \end{aligned}$$

where (a) follows from Fano's inequality with $\epsilon_n = \frac{1}{n}(H(P_e) + P_e \log_2(M-1))$. Since the channel model is memoryless, $I(X^m; Y^m) \leq \sum_{i=1}^m I(X_i; Y_i)$. Therefore,

$$\frac{1}{n} \log M \leq \sum_{i=1}^m \frac{1}{n} I(X_i; Y_i) + \epsilon_n \leq \sup_{P_X} \frac{m}{n} I(X; Y) + \epsilon_n. \quad (4)$$

Note that,

$$\begin{aligned} n &= (1-\delta)pm \times 2 + (\delta pm + (1-p)m) \times 1 \\ &= ((1-p) + \delta p + 2(1-\delta)p)m, \end{aligned} \quad (5)$$

and

$$I(X; Y) = h(Y) - h(Y|X) = h((1-\delta)p) - ph(\delta). \quad (6)$$

Substituting (5) and (6) in (4), we have

$$\frac{1}{n} \log M \leq \sup_{p \in [0,1]} \frac{h((1-\delta)p) - ph(\delta)}{(1-p) + \delta p + 2(1-\delta)p} + \epsilon_n.$$

As $n \rightarrow \infty$, $\epsilon_n \rightarrow 0$ and we have

$$C \leq \sup_{p \in [0,1]} \frac{h((1-\delta)p) - ph(\delta)}{(1-p) + \delta p + 2(1-\delta)p}.$$

Achievability: Fix a Bernoulli distribution P with parameter $p^* = \arg \sup_{p \in [0,1]} \frac{h((1-\delta)p) - ph(\delta)}{(1-p) + \delta p + 2(1-\delta)p}$, and generate a binary codebook \mathcal{C} containing 2^{mR} length m i.i.d. sequences drawn according to P , where $m = \frac{n}{(1-p) + \delta p + 2(1-\delta)p}$.

In order to send a bit '1', Alice sends a packet and then idles for one time slot. To send a bit '0', she just idles for one time slot. Thus, each message on average takes $m \times ((1-p) + \delta p + 2(1-\delta)p) = n$ time slots to be transmitted. At the same time, Bob keeps his head of queue always full. Since this is a discrete memoryless channel, by the standard typicality decoding arguments [16], the error can be kept arbitrary close to zero as long as $R \leq \max_{p \in [0,1]} I(X; Y)$. Consequently,

$$\begin{aligned} C &\geq \frac{\log 2^{m \times \max_{p \in [0,1]} I(X; Y)}}{n} \\ &\geq \max_{p \in [0,1]} \frac{h((1-\delta)p) - ph(\delta)}{(1-p) + \delta p + 2(1-\delta)p}. \end{aligned} \quad (7)$$

The achievability and converse complete the proof of the coding theorem. Therefore, $C = \sup_{p \in [0,1]} \frac{h((1-\delta)p) - ph(\delta)}{(1-p) + \delta p + 2(1-\delta)p}$. ■

Since the function $h(\cdot)$ is differentiable, to find the optimum point in the expression in Theorem 2, it suffices to take the derivative with respect to p and set it to zero. Figure 7 depicts capacity C versus the drop probability δ .

In the noisy setting, as mentioned earlier, synchronization between the encoder and decoder sides of the system may be lost. To prevent this from happening, users should utilize the fixed-length codebook design presented in Subsection IV-B2.

VI. CONCLUSION

We studied a covert queueing channel between two users sharing a round robin scheduler. An information-theoretic framework was proposed to derive the capacity of this channel in both noisy and noiseless cases. We showed that in the noiseless case, an information rate as high as 0.6942 bits per time slot is achievable in this channel. Clearly this rate of transmission can lead to significant information leakage in a system and deserves special attention in high security systems. For the noisy case, where users' packets may drop, we again analyzed the highest achievable information rate and obtained the capacity for different levels of noise. Furthermore, we propose a practical finite-length code construction, which asymptotically achieves the capacity limit.

REFERENCES

- [1] V. Anantharam and S. Verdú, "Bits through queues," *IEEE Transactions on Information Theory*, vol. 42, no. 1, pp. 4–18, 1996.
- [2] S. Cabuk, C. E. Brodley, and C. Shields, "Ip covert timing channels: design and detection," in *Proceedings of the 11th ACM Conference on Computer and Communications Security*, pp. 178–187, ACM, 2004.
- [3] S. J. Murdoch and S. Lewis, "Embedding covert channels into TCP/IP," in *International Workshop on Information Hiding*, 2005.
- [4] D. Llamas, A. Miller, and C. Allison, "An evaluation framework for the analysis of covert channels in the TCP/IP protocol suite," in *ECIW*, pp. 205–214, 2005.
- [5] M. H. Kang, I. S. Moskowitz, and D. C. Lee, "A network pump," *IEEE Transactions on Software Engineering*, vol. 22, no. 5, pp. 329–338, 1996.
- [6] X. Gong, N. Kiyavash, and P. Venkatasubramanian, "Information theoretic analysis of side channel information leakage in FCFS schedulers," in *Information Theory Proceedings (ISIT), 2011 IEEE International Symposium on*, pp. 1255–1259, IEEE, 2011.
- [7] X. Gong and N. Kiyavash, "Quantifying the information leakage in timing side channels in deterministic work-conserving schedulers," *arXiv preprint arXiv:1403.1276*, 2014.
- [8] S. Kadloor and N. Kiyavash, "Delay optimal policies offer very little privacy," in *INFOCOM, 2013 Proceedings IEEE*, pp. 2454–2462, IEEE, 2013.
- [9] S. Kadloor, N. Kiyavash, and P. Venkatasubramanian, "Mitigating timing based information leakage in shared schedulers," in *INFOCOM, 2012 Proceedings IEEE*, pp. 1044–1052, IEEE, 2012.
- [10] S. K. Gorantla, S. Kadloor, N. Kiyavash, T. P. Coleman, I. S. Moskowitz, and M. H. Kang, "Characterizing the efficacy of the nrl network pump in mitigating covert timing channels," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 1, pp. 64–75, 2012.
- [11] A. Ghassami, X. Gong, and N. Kiyavash, "Capacity limit of queueing timing channel in shared FCFS schedulers," in *2015 IEEE International Symposium on Information Theory (ISIT)*, pp. 789–793, IEEE, 2015.
- [12] R. Tahir, M. T. Khan, X. Gong, A. Ahmed, A. Ghassami, H. Kazmi, M. Caesar, F. Zaffar, and N. Kiyavash, "Sneak-peek: High speed covert channels in data center networks," in *IEEE International Conference on Computer Communications (INFOCOM)*, IEEE, 2016.
- [13] S. Kadloor, X. Gong, N. Kiyavash, T. Tezcan, and N. Borisov, "Low-cost side channel remote traffic analysis attack in packet networks," in *Communications, 2010 IEEE International Conference on*, IEEE, 2010.
- [14] S. Kadloor and N. Kiyavash, "Delay-privacy tradeoff in the design of scheduling policies," *IEEE Transactions on Information Theory*, vol. 61, no. 5, pp. 2557–2573, 2015.

- [15] <https://www.dropbox.com/s/6oli3ypufn6ggc5/paper.pdf?dl=0>.
- [16] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. John Wiley & Sons, 2012.